

Aufgabe 1

a) Classless Inter-Domain Routing (CIDR) vs. Classful Addressing

Im Gegensatz zum Classful Addressing, werden beim CIDR die Netze nicht in Klassen unterteilt. Beim Classful Addressing gibt es 3 Klassen A, B, und C von Netze, die eine unterschiedliche Anzahl von Hosts enthalten können. Netze der Klasse C können maximal 256 Hosts enthalten, die nächstgrößere Klasse B kann über 65000 Hosts aufnehmen. Da für viele Unternehmen, Firmen, etc 256 Hosts zu wenig, und 65k+ zu viel sind, gehen damit viele IP Adressen verloren, da sie nicht benutzt werden können (**geringe Effizienz**). Deswegen wurde CIDR entwickelt, die es ermöglicht die Grenze zwischen Netz-ID und Host-ID in einer IP Adressen nicht nur um mindestens 8bits zu verschieben, sondern um einen beliebigen Wert. Dazu wird zu jeder IP Adresse einen Präfix angefügt, der beschreibt, wie lang die Netz-ID ist. Dies bringt einige Vorteile mit sich:

- prefix aggregation:
Mehrere der (alten) C-Klassen Blöcke können zu einem größeren zusammengefügt werden. Somit kann die (endliche) Anzahl an IP Adressen optimaler verteilt werden und Verschwendungen vermieden werden
- route aggregation:
Mehrere IP-Adressen Blöcke können u.U. zu einem einzigen mit einer niedrigeren Präfix Nummer in den routing tables zusammengefasst werden. Damit wird die sog. routing table explosion vermieden, und die Verarbeitung im Router beschleunigt

b) Subnetting

Netz: 192.168.130.0 = 11000000 10011110 10000010 00000000
Subnet Maske: 255.255.255.224 = 11111111 11111111 11111111 **11100000**

Drei Bits sind für die Subnets-ID reserviert und **5** für die Host-ID.

Das ermöglicht also $2^5 = 32$ Hosts pro Subnet, wobei man die Hosts 0 (lauter 0) und -1 (lauter 1) abziehen muss, also **30 Hosts pro Subnet**.

Mit **3** Bits können $2^3 = 8$ **Subnets** adressiert werden. Eigentlich müsste man die Subnets 0 und -1 subtrahieren, da sie streng genommen reserviert sind (und somit die Anzahl der Subnets auf 6 bringen), doch dies wird in der Praxis nie gemacht. Im Folgenden werden also diese Subnets mitberechnet, falls man diese nicht mithaben möchte, so muss man die mit (*) versehenen Lösungen ignorieren.

Tabelle

ID	Adresse	Bereich	Broadcast
Subnet 1 (*)	192.168.130.0	192.168.130.1 – 192.168.130.30	192.168.130.31
Subnet 2	192.168.130.32	192.168.130.33 – 192.168.130.62	192.168.130.63
Subnet 3	192.168.130.64	192.168.130.65 – 192.168.130.94	192.168.130.95
Subnet 4	192.168.130.96	192.168.130.97 – 192.168.130.126	192.168.130.127
Subnet 5	192.168.130.128	192.168.130.129 – 192.168.130.158	192.168.130.159
Subnet 6	192.168.130.160	192.168.130.161 – 192.168.130.190	192.168.130.191
Subnet 7	192.168.130.192	192.168.130.193 – 192.168.130.222	192.168.130.223
Subnet 8 (*)	192.168.130.224	192.168.130.225 – 192.168.130.254	192.168.130.255

Beispiel IPs:

192.168.130.96
192.168.130.67
192.168.130.93
192.168.130.199
192.168.130.222
192.168.130.255

Beispiel 1:

192.168.130.96 = 11000000 10011110 10000010 01100000

255.255.255.244 = 11111111 11111111 11111111 11100000

AND Verknüpfung ergibt:

11000000 10011110 10000010 01100000 = 192.168.130.96, was **genau Subnet 4** ist.

Beispiel 2:

192.168.130.67 = 11000000 10011110 10000010 01000011

255.255.255.244 = 11111111 11111111 11111111 11100000

AND Verknüpfung ergibt:

11000000 10011110 10000010 01000000 = 192.168.130.64, also gehört 192.168.130.67 zu

Subnet 3, und ist **Host nr. 3**

Beispiel 3:

192.168.130.93 = 11000000 10011110 10000010 01011101

255.255.255.244 = 11111111 11111111 11111111 11100000

AND Verknüpfung ergibt:

11000000 10011110 10000010 01000000 = 192.168.130.64, also gehört 192.168.130.93 zu

Subnet 3, und ist **Host nr. 29**

Beispiel 4:

192.168.130.199 = 11000000 10011110 10000010 11000111

255.255.255.244 = 11111111 11111111 11111111 11100000

AND Verknüpfung ergibt:

11000000 10011110 10000010 11000000 = 192.168.130.192, also gehört 192.168.130.199

zu **Subnet 7**, und ist **Host nr. 7**

Beispiel 5:

192.168.130.222 = 11000000 10011110 10000010 11011110

255.255.255.244 = 11111111 11111111 11111111 11100000

AND Verknüpfung ergibt:

11000000 10011110 10000010 11000000 = 192.168.130.192, also gehört 192.168.130.222

zu **Subnet 7**, und ist **Host nr. 30**

Beispiel 2:

192.168.130.255 = 11000000 10011110 10000010 11111111

255.255.255.244 = 11111111 11111111 11111111 11100000

AND Verknüpfung ergibt:

11000000 10011110 10000010 11100000 = 192.168.130.224, also gehört 192.168.130.255

zu **Subnet 8**, und ist an alle Hosts gerichtet (**Broadcast**)

Aufgabe 2

Durchschnittliche Streckenanzahl = Summe aller Strecken / Anzahl von Nachrichten

a) vollständig vermaschte Rechnernetze

Alle Rechner sind mit allen anderen verbunden, d.h. eine Nachricht kann vom Sender zum Empfänger direkt geschickt werden:

Weg 1: 1 Strecke

Weg 2: 1 Strecke

...

Weg N: 1 Strecke

Summe der Strecken: $1 * N = N$

Anzahl der Nachrichten: N

$N/N = 1 = \text{Durchschnittliche Streckenanzahl}$

b) Stern mit Rechner in der Mitte

Ein Rechner steht in der Mitte und kann jeden anderen Rechner direkt erreichen.

Alle anderen Rechner können diesen auch direkt erreichen und alle anderen in 2 Teilstrecken, nämlich durch den Weg: Start – Zentralrechner – Ziel.

Da alle Rechner zu jedem anderen Rechner mit gleicher Wahrscheinlichkeit Nachrichten schicken und auch in gleichen Abständen, gilt folgendes:

Jeder Rechner schickt zu einem Zielrechner mit der Wahrscheinlichkeit $1/(N-1)$ (-1 weil Rechner im Bsp nicht sich selbst Nachrichten schicken können). Pro Rechner gibt es $N-1$ Nachrichten die ankommen könnten. D.h. jeder Rechner bekommt im Schnitt

$1/(N-1) * (N-1) = 1$ Nachrichten.

Also gilt:

Weg 1: 1 (der Weg der Nachricht die vom zentralen Rechner ausgeht)

Weg 2: 1 (der Weg der Nachricht zum zentralen Rechner)

Alle anderen Wege: 2

Summe der Strecken: $1 + 1 + 2(N-2) = 2*(N-1)$

Anzahl der Nachrichten: N

$2*(N-1) / N = \text{Durchschnittliche Streckenanzahl}$

c) Stern ohne Rechner in der Mitte

Alle Rechner können jeden anderen Rechner in 2 Teilstrecken erreichen, nämlich durch den Weg Sender – Mitte – Empfänger.

Also gilt:

Weg 1: 2 Strecken

Weg 2: 2 Strecken

...

Weg N: 2 Strecken

Summe der Strecken: $2 * N = N$

Anzahl der Nachrichten: N

$2*N/N = 2 = \text{Durchschnittliche Streckenanzahl}$

Aufgabe 3

1. Flache Adressierung z.B. Ethernet Adressen:

- global eindeutig
- keine Struktur (kaum Hinweise für Routing-Protokolle)
- schlecht skalierbar (routing table explosion)

2. Hierarchische Adressierung z.B. IP:

- vermeidet Probleme der flachen Adressierung
- IP Adressen: Netz und Host Teil (relativ logische Struktur für ein Internetwork -> Informationen für Router)
- jeder Netzbereich besitzt einen Border-Router oder Gateway als Verbindung zu anderen Netzbereichen
- dieser Router kann ein Single-Point-of-Failure sein

Aufgabe 4

Internet Modell

Anwendung
Transport
Netz
Netzzugang

OSI/ISO Modell

7	Application Layer Anwendungsschicht
6	Presentation Layer Darstellungsschicht
5	Session Layer Sitzungsschicht
4	Transport Layer Transportschicht
3	Network Layer Vermittlungsschicht
2	Data Link Layer Sicherungsschicht
1	Physical Layer Bitübertragungsschicht

- **Physical Layer:**
Zuständig für die Übertragung einzelner Bits. Legt fest wie Bits aussehen (Signaldauer, Volt, usw), wie die Verbindung aufgenommen und wieder unterbrochen wird, wer wann senden darf, ob gleichzeitig oder nacheinander gesendet werden darf, usw.
- **Data Link Layer:**
Wandelt bits-Folge in einer sauberen (fehlerfreien) Form um, teilt evtl. zu große Pakete in mehrere data-frames und regelt was passiert wenn der Sender schneller senden als der Empfänger empfangen kann.
- **Network Layer:**
Kontrolliert die Operationen des Subnets, kümmert sich also ums routing, oder löst evtl. Kompatibilitätsprobleme zwischen verschiedene Netze, falls Protokolle sich unterscheiden sollten, usw. (IP)
- **Transport Layer:**
Nimmt Datenpakete von den höheren Schichten an, kümmert sich darüber, dass sie wirklich ankommen, und wählt den Dienst. Stellt sicher, dass bei Änderungen der Hardware (die unteren Schichten) die oberen Schichten nicht geändert werden müssen (TCP)
- **Session Layer:**
Ermöglicht es Nutzern auch verschiedenen Rechner Sitzungen miteinander aufzustellen und regelt diese.
- **Presentation Layer:**
Beschäftigt sich mit Syntax und Semantik der erhaltenen Daten und nicht mit deren physischen Übertragung. Notwendig da verschiedene Maschinen verschieden Darstellungen gleicher Daten verwenden.
- **Application Layer:**
Enthält eine Reihe von wichtigen Protokolle für den Nutzer, wie z.B. HTTP